

SISTEMA DE ENSINO À DISTÂNCIA - UNOPAR

PROGRAMA DE GRADUAÇÃO

UNOPAR

PRODUÇÃO TEXTUAL INTERDICÍPLINAR

Cidade

2013

Teu Nome

PRODUÇÃO TEXTUAL INTERDICIPLINAR

Trabalho apresentado as disciplinas de Linguagem E Programação e Estruturas de Dados, Banco de Dados II, Análise de Sistemas II e Sistemas Operacionais. do 3º. Semestre do Curso de Análise e Desenvolvimento de Sistemas da Universidade Norte do Paraná - UNOPAR

Profs.: Polyanna P. Gomes
Adriane Loper
Roberto Nishimura
Merris Mozer

Cidade

2013

SUMÁRIO

1. Capa.....	01,02
2.Introdução.....	03
3.Objetivo.....	04
4.Desenvolvimento.....	05
4.1.Propriedades Do ACID.....	05
4.2.Sistemas Operacionais.....	07
4.3.Estrutura De Dados Fila E Pilha.....	09
4.4.Diagrama De Caso E De Uso.....	11
4.5.Humanismo E Ética No Trabalho.....	19
5. Conclusão.....	22
6. Referências.....	23

2. INTRODUÇÃO

Não podemos mais negar a importância da interdisciplinaridade nos dias de hoje em sala de aula, cada vez mais temos que trabalhar em conjunto – professores e alunos de todas as disciplinas por isso torna-se necessário o trabalho por projetos e integrando todas as disciplinas.

Nesse portfólio, por exemplo: elaborar um trabalho sobre todas as disciplinas dos 3 semestres ajuda a ligar o conteúdo à prática de uma maneira mais eficiente e eficaz.

3. OBJETIVO

O objetivo é alcançar aprimoramento nos sistemas, na busca de crescimento e aprofundamento no mercado da Tecnologia, trazendo para si as melhorias alcançadas, desenvolvendo Programas os quais auxiliem e facilitem o dia a dia dos nossos clientes tendo em vista que; para aprimorar cada vez mais nosso meio de trabalho devemos obter grande conhecimento, favorecimento na Ética e disciplina na Sociedade, sabendo que desta forma a responsabilidade aumenta tanto na postura de assumir como um Analista, em sua busca de satisfazer não somente o cliente alvo e sim crescer neste mundo da Tecnologia um tanto devastador. Por isso cabe ao profissional obter mais conhecimento, postura e conduta no seu meio de trabalho e assim aperfeiçoar seus conhecimentos naquilo que ele deseja aprofundar-se seja em quaisquer linguagens de programação e ou Sistemas Operacionais.

4. DESENVOLVIMENTO

Todo e quaisquer Sistemas, deve-se ter uma boa fase de desenvolvimento, estas fases implica-se em suas aplicações, estas aplicações auxiliam aos Analistas e Desenvolvedores de Sistemas. Há alguns conceitos, os quais irão abordar aqui:

4.1. Propriedades Do ACID

ACID (acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade), é um conceito utilizado para caracterizar uma transação.

Atomicidade

Trata o trabalho como parte indivisível (atômico). A transação deve ter todas as suas operações executadas em caso de sucesso ou nenhum resultado de alguma operação refletida sobre a base de dados em caso de falha. Ou seja, após o término de uma transação (commit ou abort), a base de dados não deve refletir resultados parciais da transação.

Exemplos:

Ou todo o trabalho é feito, ou nada é feito.

Em uma transferência de valores entre contas bancárias, é necessário que, da conta origem seja retirado um valor X e na conta destino seja somado o mesmo valor X. As duas operações devem ser completadas sem que qualquer erro aconteça, caso contrário todas as alterações feitas nessa operação de transferência devem ser desfeitas;

CONSISTÊNCIA

Regras de integridade dos dados são asseguradas, ou seja, as transações não podem quebrar as regras do Banco de Dados. no contexto de

banco de dados, o termo refere-se a transações que não violem nenhuma restrição de integridade durante a sua execução. Se a transação deixa a base de dados em um estado incorreto, é abortado e é reportado um erro.

Isolamento

Tudo se parece como se o trabalho estivesse isolado. O resultado de uma transação executada concorrentemente a outra deve ser o mesmo que o de sua execução de forma isolada. Operações exteriores a uma dada transação jamais verão esta transação em estados intermediários.

Ex: Duas transações executadas juntas.

Transação Y altera o nome do cliente.

Transação X altera o nome do mesmo cliente;

Ao fazer insert, update ou delete enquanto a transação estiver em andamento todos os dados da tabela serão alocados para aquela transação, portanto a transação X só poderá ser executada após a conclusão da transação Y. Esta escolha é feita pelo banco de dados e ele executa primeiro uma transação depois a outra.

Duas transações só podem ser simultâneas se elas não alterarem os mesmos dados

DURABILIDADE

Os efeitos de uma transação em caso de sucesso (commit) são permanentes mesmo em presença de falhas. Significa que os resultados de uma transação são permanentes e podem ser desfeitos somente por uma transação subsequente.

4.2. Sistemas Operacionais

Numa Locadora de Livros podemos usar tanto o Sistema Linux como o Windows, dando este exemplo:

Nossa proposta foi implementa um sistema de locadora de Livros. Onde cada Livro possui um número. Para cada livro, é necessário saber seu título e sua categoria (comédia, drama, romance, ...). Cada livro recebe um identificador próprio. Para cada livro é controlado que categoria ele contém. Para cada categoria há pelo menos um livro e cada livro contém somente uma categoria. Alguns poucos necessitam de 2 livros. Os clientes podem desejar encontrar livros por categoria e também livros estrelados por autores prediletos. Por isso, é necessário manter informação sobre autores que estrelam cada livro.

A locadora possui muitos clientes cadastrados. Somente clientes cadastrados podem alugar livros. Para cada cliente é necessário saber seu nome, telefone e endereço. Além disso, cada cliente recebe um número de associado. Por fim, deseja-se saber que livros cada cliente tem emprestado no momento e qual o histórico de aluguéis efetuado. Um cliente pode ter vários livros alugados em um instante do tempo.

RECURSOS UTILIZADOS:

Sistema de gerenciamento de banco de dados (SGDB) que possui uma grande compatibilidade com várias linguagens de programação como Java, Delphi, PHP, etc.. Possui uma grande vantagem de ter o código aberto, um bom desempenho, e pode funcionar com vários sistemas operacionais.

Atualmente o MySQL é o banco de dados de código aberto mais popular do mundo, com mais de 6 milhões de instalações entre Websites, datawarehouse, aplicações comerciais e outra mais. Usuários como Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics, e Texas Instruments usam o MySQL em aplicações de missão crítica.

Características do MySQL

- Rapidez
- Facilidade de instalar, usar e administrar.
- Confiabilidade
- Sistema de aviso e informação através de pacote de suporte – MySQL Network

Uma linguagem de programação orientada a objetos utilizada para desenvolvimento web, que permite-nos colocar os seus pequenos fragmentos de código dentro da página HTML e realizar determinadas tarefas de uma forma fácil e eficaz sem ter de implementar programas programados na íntegra numa linguagem diferente de HTML.

Ferramenta escolhida para podermos simular um servidor e permitir a configuração do PHP, essa ferramenta possui três componentes indispensáveis como o Apache, servidor de páginas web, MySQL, banco de dados de código livre, e o PHP, linguagem para web para acesso ao banco de dados.

Ferramenta de modelagem de banco de dados que facilita e simplifica a geração da engenharia reversa. Geramos o script para podermos utilizá-lo no banco de dados MySQL.

DESENVOLVIMENTO DO SISTEMA

Inicialmente dialogamos sobre os diferentes sistemas que poderíamos fazer, até chegarmos ao sistema descrito anteriormente. No outro dia, foi escolhido a linguagem de implementação que seria utilizada, PHP.

Usando o Power Architect, desenhamos o diagrama.

Entidade-Relacionamento

A partir deste modelo foi gerado o script para a geração do Banco de Dados, veja abaixo o resultado, que já inclui os comandos que foram dados para colocar informação nas tabelas e as views que foram parte do sistema:

CONCLUSÃO:

Finalizando este trabalho, concluímos que o saldo foi positivo, ou seja, o desafio de programar em php e a utilizar de um banco de dados nunca antes testado trouxe resultados satisfatórios. Realmente o banco é muito eficiente, fácil de usar e possui funcionalidades muito importantes, que possibilitam consultas mais complexas, como views, por exemplo. A linguagem utilizada possui recursos variados, o que permitiu uma boa ligação com o banco de informações, mysql. Assim, conseguimos ver na prática, como as consultas estudadas em aula são aplicadas no dia-a-dia e que ajudam e muito, a vida do usuário e do próprio gerente de negócio.

4.3. Estrutura De Dados Fila E Pilha

Estrutura de dados:

Sobre as estruturas de dados veremos duas formas de estruturas a Fila (FIFO) e a pilha (LIFO).

Fila - Uma fila é uma estrutura de dados que admite inserção de novos elementos e remoção de elementos antigos. Mais especificamente, uma fila (= que) é uma estrutura sujeita à seguinte regra de operação: sempre que houver uma remoção, o elemento removido é o que está na estrutura há mais tempo.

Em outras palavras, o primeiro objeto inserido na fila é também o primeiro a ser removido. Essa política é conhecida pela sigla FIFO (= First-In-First-Out).

Pilha - Uma pilha é uma das várias estruturas de dados que admitem remoção de elementos e inserção de novos elementos. Mais especificamente, uma pilha (= stack) é uma estrutura sujeita à seguinte regra de operação: sempre que houver uma remoção, o elemento removido é o que está na estrutura há menos tempo.

Em outras palavras, o primeiro objeto a ser inserido na pilha é o último a ser removido. Essa política é conhecida pela sigla LIFO (= Last-In-First-Out).

A. Listas Simplesmente Encadeadas E Duplamente Encadeadas.

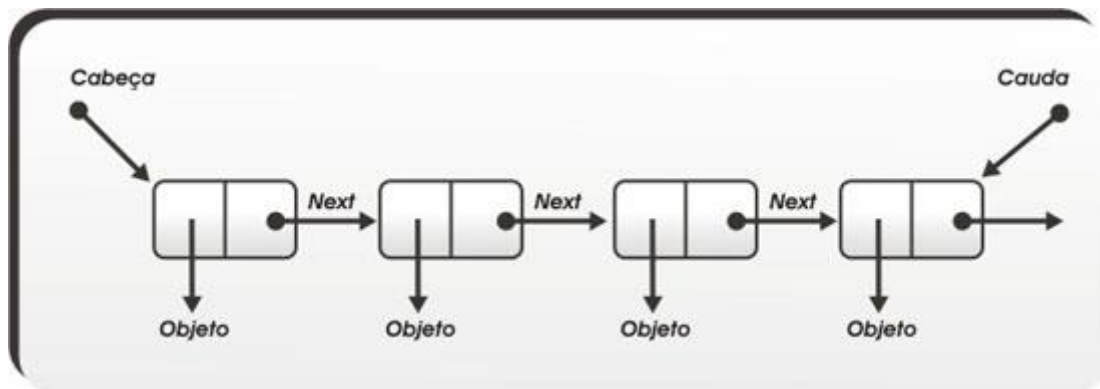
Lista simplesmente encadeada é uma vantajosa alternativa na implantação de arrays. Arrays implica que o tamanho de N elementos devem ser fixados a priori.

Uma alternativa para esta desvantagem é o uso de lista encadeada, que permite a criação de arrays sem um número de N elementos fixo.

Uma lista é formada por nós. Nó é um objeto composto que armazena uma referência para um elemento ou objeto qualquer e uma referência next (próximo) que permite fazer a ligação com o próximo nó da lista até que o último aponte para uma região de memória nula (null).

O primeiro e os últimos nós de uma lista são chamados respectivamente de cabeça (referência head) e cauda (referência tail) da lista.

Exemplo de Lista Simplesmente Encadeada



Uma lista duplamente ligada (ou lista duplamente encadeada) é uma extensão da lista simplesmente ligada (ou lista simplesmente encadeada).

Numa lista cada elemento, ou nó, é composto normalmente por uma variável que guarda a informação (Objeto, inteiro, cadeia de caracteres, etc.) e dois ponteiros (referências a endereços de memória) que permitem a ligação entre os vários nós desta lista. Este tipo de lista é conhecido por "Duplamente ligada" ou "Duplamente encadeada" exatamente pelo fato de possuir duas variáveis de controle (ponteiros) ao contrário da lista simplesmente ligada que possui somente um, o qual aponta para o próximo elemento da lista.

A função destas variáveis é guardar o endereço de memória do nó anterior e do nó posterior, identificados normalmente como "prev" ou "previous" e "next". Com estas estruturas podemos realizar diversas tarefas que seriam impossíveis ou muito dispendiosas com uma lista simplesmente encadeada.

No modelo mais simples deste tipo de lista, ao criar a lista o primeiro nó tem seu ponteiro "previous" apontando sempre para nulo e o último nó com seu "next" apontando para nulo. Este modelo não é muito confiável, já que não há um controle efetivo para saber quem é o primeiro e quem é o último elemento, já que a única maneira de extrair tal informação é verificar quem possui o "prev" ou o "next" nulo.

Existem várias ramificações da lista duplamente encadeada, e muitas delas servem também para a lista simplesmente encadeada. Aqui

temos alguns exemplos:

Lista duplamente encadeada com sentinelas: Neste modelo de lista possuímos dois Nós estáticos a cabeça da lista (head) e o fim da lista(tail). O elemento prev (anterior) do nó head aponta sempre para nulo enquanto no nó tail quem aponta para nulo é o next(próximo).

Vantagens: Maior facilidade de controle da lista, maior confiabilidade e menor risco de perda acidental da lista.

Desvantagens: Maior gasto de espaço em disco (2 nós a mais).

Lista duplamente encadeada circular: Neste modelo de lista possuímos apenas um sentinela. Esta lista é conhecida como circular pois o sentinela aponta para o primeiro elemento da lista e o último elemento da lista aponta para o sentinela, formando assim um círculo lógico.

Vantagens: Economia de espaço em disco (1 nó a menos que a lista duplamente encadeada com sentinelas), maior confiabilidade em relação ao modelo comum.

Desvantagens: Maior complexidade nos algoritmos.

4.4. Diagrama De Caso E De Uso.

Casos de Uso

Diagrama de Casos de Uso

Objetivo

O Diagrama de *Casos de Uso* tem o objetivo de auxiliar a comunicação entre os analistas e o cliente.

Um diagrama de Caso de Uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário.

O cliente deve ver no diagrama de Casos de Uso as principais funcionalidades de seu sistema.

Notação

O diagrama de Caso de Uso é representado por:

- atores;
- casos de uso;
- relacionamentos entre estes elementos.

Estes relacionamentos podem ser:

- associações entre atores e casos de uso;
- generalizações entre os atores;
- generalizações, *extends* e *includes* entre os casos de uso.

casos de uso podem opcionalmente estar envolvidos por um retângulo que representa os limites do sistema.

Em maiores detalhes:

• Atores



Ator

Um ator é representado por um boneco e um rótulo com o nome do ator. Um ator é um usuário do sistema, que pode ser um usuário humano ou um outro sistema computacional.

• Caso de uso



Caso de Uso

Um *caso de uso* é representado por uma elipse e um rótulo com o nome do *caso de uso*. Um *caso de uso* define uma grande função do sistema. A implicação é que uma função pode ser estruturada em outras funções e, portanto, um *caso de uso* pode ser estruturado.

• Relacionamentos

- Ajudam a descrever *casos de uso*
- Entre um ator e um *caso de uso*
 - Associação



Ator

Caso de Uso

Define uma funcionalidade do sistema do ponto de vista do usuário.

- Entre atores
 - Generalização

Objetivo

Descrever os vários tipos de objetos no sistema e o relacionamento entre eles.

Perspectivas

Um diagrama de classes pode oferecer três perspectivas, cada uma para um tipo de usuário diferente. São elas:

- Conceitos ou Entidades (exemplo; mais_exemplos)
 - o Representa os conceitos do domínio em estudo.
 - o Perspectiva destinada ao cliente.
- Classes (exemplo)
 - o Tem foco nas principais interfaces da arquitetura, nos principais métodos, e não como eles irão ser implementados.
 - o Perspectiva destinada as pessoas que não precisam saber detalhes de desenvolvimento, tais como gerentes de projeto.
- Classes de Software (exemplo)
 - o Aborda vários detalhes de implementação, tais como navegabilidade, tipo dos atributos, etc.
 - o Perspectiva destinada ao time de desenvolvimento.

Um diagrama de classes contém:

- Entidades
- Relacionamentos
- Exemplo contendo as notações

Exemplo Completo

Especificação do exemplo

Solução 1

Solução 2 (para quem tiver mais curiosidade)

Entidades

Classe

Classe Concreta

Uma classe é representada na forma de um retângulo, contendo duas linhas que separam 3 partes. A primeira contém no nome da classe, a segunda os atributos da classe e a última os métodos da mesma.

Classe Abstrata

A representação de uma classe abstrata em UML é quase igual à representação de uma classe concreta, a única diferença é o estilo da fonte do nome da classe, que, neste caso, está em *itálico*.

Interface

Perspectivas:

Conceitual

Apenas classes são utilizadas. Neste tipo de perspectiva, uma classe é interpretada como um conceito. Apenas atributos são utilizados.

Especificação

Tanto classes como interfaces são utilizados neste tipo de perspectiva. O foco consiste em mostrar as principais interfaces e classes juntamente com seus métodos.

Não é necessário mostrar todos os métodos, pois o objetivo deste diagrama nesta perspectiva é prover uma maior entendimento da arquitetura do software a nível de interfaces.

Implementação

Nesta perspectiva, vários detalhes de implementação podem ser abordados, tais como:

Visibilidade de atributos e métodos;

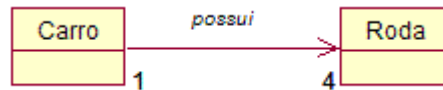
Parâmetros de cada método, inclusive o tipo de cada um;

Tipos dos atributos e dos valores de retorno de cada método.

Relacionamentos

- Papel

Descreve o relacionamento.



- Multiplicidade (utilizado em todas as perspectivas de forma uniforme)

Notações possíveis

<i>Tipos</i>	<i>Significa</i>
0..1	Zero ou uma instância. A notação n..m indica n para m instâncias.
0..* ou *	Não existe limite para o número de instâncias.
1	Exatamente uma instância.
1..*	Ao menos uma instância.

Exemplos:



- Associação (utilizado em todas as perspectivas)

Representação Gráfica

Associação

Perspectiva:

- Conceitual

Define um relacionamento entre duas entidades conceituais do sistema.

- Especificação

Diagramas de Interação são modelos que descrevem como grupo de objetos

colaboram em um determinado comportamento.

Um diagrama de interação captura o comportamento entre objetos dentro um único use case.

Utiliza-se o diagrama de atividade para representar o comportamento de objetos entre vários use cases.

Tipos:

Diagrama de Sequencia

Diagrama de Colaboração

Diagramas de Interação

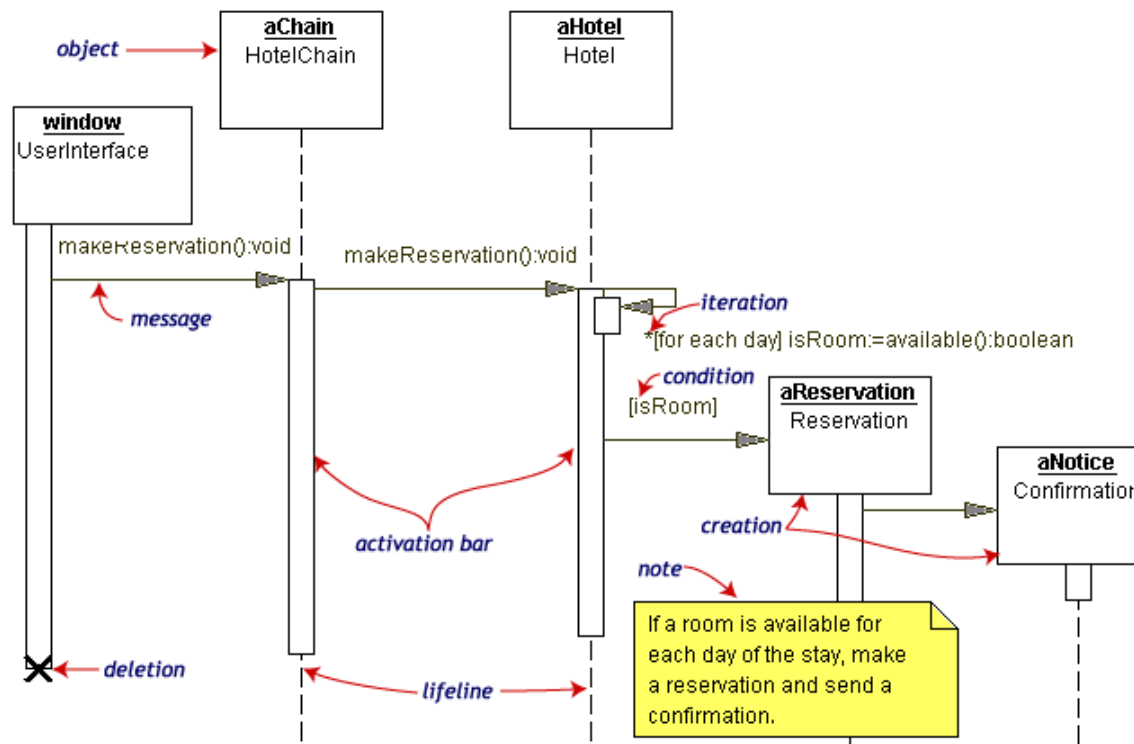
Diagramas de Sequencia

Consiste em um diagrama que tem o objetivo de mostrar como as mensagens entre os objetos são trocadas no decorrer do tempo para a realização de uma operação.

Em um diagrama de sequencia, os seguintes elementos podem ser encontrados:

- Linhas verticais representando o tempo de vida de um objeto (lifeline);
- Estas linhas verticais são preenchidas por barras verticais que indicam exatamente quando um objeto passou a existir. Quando um objeto desaparece, existe um "X" na parte inferior da barra;
- Linhas horizontais ou diagonais representando mensagens trocadas entre objetos. Estas linhas são acompanhadas de um rótulo que contém o nome da mensagem e, opcionalmente, os parâmetros da mesma. Observe que também podem existir mensagens enviadas para o mesmo objeto, representando uma iteração;
- Uma condição é representada por uma mensagem cujo rótulo é envolvido por colchetes;
- Mensagens de retorno são representadas por linhas horizontais tracejadas. Este tipo de mensagem não é frequentemente representada nos diagramas, muitas vezes porque sua utilização leva a um grande número de setas no diagrama, atrapalhando o entendimento do mesmo. Este tipo de mensagem só deve ser mostrada quando for fundamental para a clareza do diagrama.

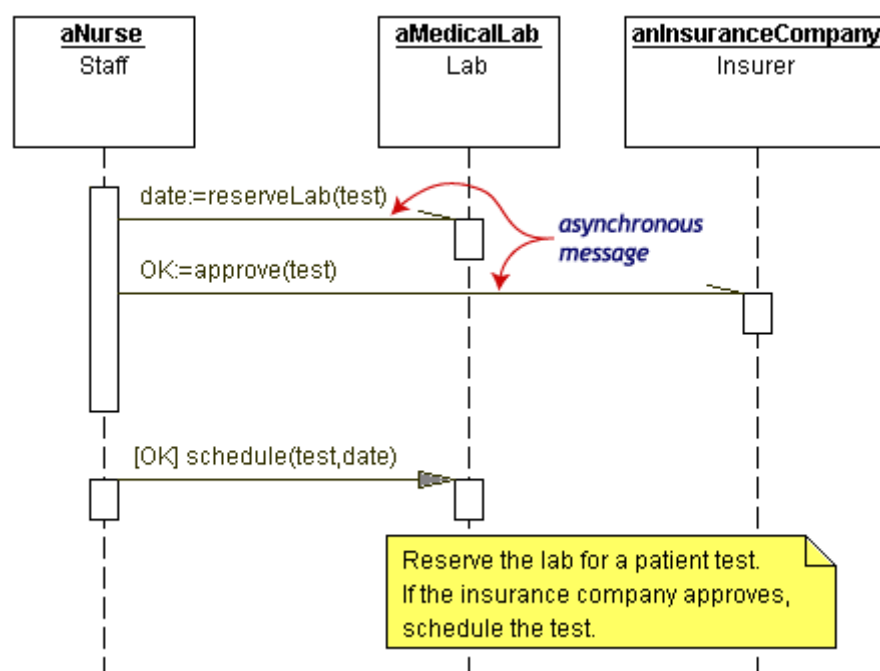
Observe a figura abaixo.



Representado processos concorrentes

Este tipo de diagrama também permite representar mensagens concorrentes assíncronas (mensagens que são processadas em paralelo sem um tempo definido para a sua realização).

Exemplo:



Diagramas de Interação

Diagramas de Colaboração

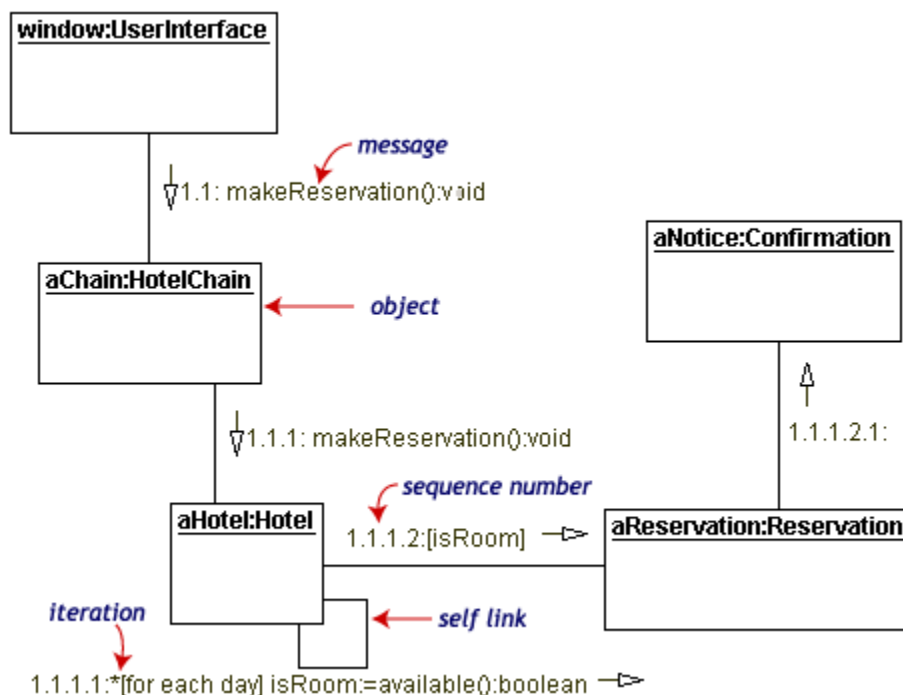
A grande diferença entre um diagrama de colaboração e um de seqüência consiste no fato de que o tempo não é mais representado por linhas verticais, mas sim através de uma numeração, que pode ser de duas formas:

- simples (1,2,3,...)
- composta (1.1, 1.2, 1.2.1, ...)

Um objeto é representado como um retângulo, contendo no seu interior um rótulo, que informa o nome do objeto e o nome da classe, separados por dois pontos. Detalhe: ambos podem ser omitidos.

A troca de mensagens entre os objetos segue o mesmo padrão que o apresentado nos diagramas de seqüência.

Veja o exemplos abaixo:



4.5 Humanismo E Ética no Trabalho.

A - Paola Antonelli é curadora de arquitetura e design do MoMa, museu de arte moderna de Nova York, e concedeu uma entrevista para o The Creators Project abordando conceitos de humanização da tecnologia relacionados ao seu projeto chamado Talk To Me, que será lançado no verão americano.

Ela defende que a tecnologia sempre existiu em diferentes momentos e intensidades, ela sempre foi o meio que possibilitou as pessoas conversar com os objetos à nossa volta e vice-versa, mudando a forma como as pessoas vivem e como os profissionais trabalham. Hoje, com um alto consumo de bens tecnológicos, as pessoas condenam a tecnologia como um problema, algo que desumaniza as pessoas.

O uso excessivo realmente pode ser prejudicial, mas não podemos negar a tecnologia nas nossas vidas, não podemos mais viver sem ela. Temos que, sim, incentivá-la para um uso que gere desenvolvimento, como na visão de Paola: “Many people think that technology is a problem in that it dehumanizes people. And, instead, I think it’s a great thing because it humanizes objects”.

Essa abordagem de humanizar objetos nos permite uma reflexão interessante de qual é o papel da tecnologia no nosso dia-a-dia. Alguns exemplos de obras que serão expostas no projeto são: The Lost Tribes of New York City, o Tenori-On, I want You To Want Me e PIG 05049. Esses projetos demonstram um pouco do que significa humanizar a tecnologia e eles são excelentes referências, não apenas como linguagem estética e artística, mas de como isso poderia ser aplicado para marcas. Afinal, como poderíamos humanizar as marcas?

B - As pessoas estão se relacionando cada vez mais através de meios tecnológicos eletrônicos e digitais. Com a evolução das tecnologias de comunicação, iniciando pelo telégrafo de Samuel Morse, e, posteriormente, o telefone de Alexander Graham Bell, o homem vivenciou verdadeiras revoluções em seu modo de comunicação. Passando pelas tecnologias de radiocomunicação, satélites de comunicação, cabos ópticos, computador, Internet e telefonia celular, e, atualmente, integração entre muitas destas tecnologias, as pessoas podem experimentar um nível de comunicação sem precedentes. Após o advento da Internet, redes sociais foram criadas com os mais variados fins, porém todas têm um objetivo comum, proporcionar o relacionamento entre duas ou mais pessoas. Supõe-se que o medo de contatos presenciais, devido à violência que cresce a cada dia, agilidade na comunicação proporcionada pelos dispositivos eletrônicos e digitais, a inserção social que é auxiliada pelas redes sociais, já que tais redes podem diminuir o impacto de fatores emocionais como timidez, auto-estima e preconceitos

podem influenciar na preferência pela utilização de tais meios de comunicação. Além do fato de que quando utiliza a Internet, o jovem pode experimentar uma oportunidade lúdica, seja através de jogos ou representação de personagens, pode buscar também uma identidade com outras pessoas, tendo inclusive oportunidade de dialogar sobre os mais variados temas. A questão relevante nessa pesquisa é verificar a motivação dos jovens quanto à utilização destas tecnologias eletrônicas e digitais para se comunicarem como mediadores tecnológicos em seus relacionamentos interpessoais.

C - Uma das questões que se levanta com grande frequência nesse início do Terceiro Milênio é a da postura ética no uso da tecnologia. Quando o assunto é esse, não importa que estejamos falando de áreas completamente distintas da atuação humana, como as de medicina, jornalismo, física ou marketing. Em qualquer uma delas, o emprego da tecnologia sempre levanta uma indagação a respeito de quais limites devem ser respeitados e de quais devem ser superados.

Mas será que essa é uma problemática exclusivamente contemporânea? Viveríamos nós em um período assim especial, ao qual não se aplicam os preceitos herdados dos antigos, medievais e modernos? Ou o caso seria outro?

A proposta do presente artigo é a de sugerir uma rápida e resumida reflexão dirigida a respeito do assunto, sem a pretensão de oferecer soluções nem análises profundas da questão apresentada.

Ao passarmos a história da humanidade em rápida revista, encontramos indícios claros de que todos os desenvolvimentos tecnológicos levaram a humanidade a situações merecedoras de maior atenção no campo ético, mesmo que não se observe nenhuma revisão ética de fato.

Como exemplo disso, podemos lembrar passagens históricas significativas, em que o uso de novas tecnologias, principalmente nas guerras, levou a uma mudança no equilíbrio do poder. Recuando uns dois mil e quinhentos anos no tempo, encontramos a prática utilizada pelos bárbaros de lançar "bombas" de veneno sobre os depósitos de sementes das cidades sitiadas, para minar a resistência à invasão. Tal estratégia empregava o uso de diversas técnicas, como as de extrair e combinar materiais para se obter o veneno, ou ainda a tecnologia envolvida na construção das armas de arremesso utilizadas, desde fundas até catapultas.

Ao longo da Idade Média, surgiram novas armas de destruição em grande escala, capazes de eliminar um grande número de inimigos em pouco tempo,

sustentando a idéia de que não basta possuir um exército maior, ou melhor, mas que também é necessário haver superioridade tecnológica.

Esses exemplos nos inspiram a fazer uma pergunta: o uso dessas tecnologias bélicas foi ético? Talvez seja difícil responder ao certo. Há argumentos tanto contrários quanto favoráveis ao uso da tecnologia para "desequilibrar" uma guerra. Em geral, a opinião do lado vencedor é a que prevalece e, também em geral, foi esse lado que utilizou uma tecnologia superior.

Podemos encontrar exemplos ainda mais claros no final da Idade Média e no início da Idade Moderna, quando as grandes navegações levaram os europeus a encontrar povos tecnologicamente tão rudes que a superioridade de suas armas lhes permitiu colonizar novas terras. Em seguida, a Revolução Científica trouxe à luz uma série de conhecimentos que se reverteram em tecnologia e em novas possibilidades de "domínio" sobre a natureza. Mais uma vez, questionar tais passagens quanto à ética é uma árdua tarefa, pois o "lado vencedor" foi aquele que deixou sua versão gravada para o futuro.

E isso tende a se repetir indefinidamente na história.

5. CONCLUSÃO

O que é então a tecnologia? Tecnologia é poder. É o poder de controlar, de criar, de aplicar a razão sobre a natureza, sobre as pessoas. Quanto mais avançada a tecnologia, maior o controle e mais amplo o poder. Em uma sociedade absolutamente tecnológica, o poder seria absoluto? Talvez. Isso soa assustador, mas o mais importante não é saber "quem" teria acesso a esse poder, mas sim conhecer quais os fatores que delimitam esse poder, uma vez que quando se fala em "poder absoluto", está-se referindo a algo acima de qualquer lei.

A proposta mais coerente seria a de esse poder se submeter apenas aos princípios éticos, independentemente da área de atuação. Diferentemente dos antigos, que puderam se abster de uma manifestação mais premente da ética, dado o estágio tecnológico em que se encontravam a geração atual não pode mais se permitir tal indulto. Tal permissividade seria uma espécie de "suicídio" por "asfixia tecnológica", já que a falta de ética poderia levar a humanidade a consequências desastrosas.

A discussão das questões éticas nas aplicações tecnológicas deve permear o diálogo científico nos anos vindouros. Desse embate surgirá o mapeamento de nosso futuro. "Quem viver verá".

6. Referências

- ABBAGNANO, N. Dicionário de Filosofia. Trad. de Alfredo Bosi (Org.). São Paulo: Martins Fontes, 1999.
- ARISTÓTELES. Ética a Nicômaco. Trad. de Leonel Vallandro e Gerd Bornheim. "Coleção: Os Pensadores". São Paulo: Abril Cultural, 1984.
- COULANGES, F. A Cidade Antiga. Trad. de Fernando de Aguiar. São Paulo: Martins Fontes, 1998.
- FOUREZ, G. A Construção das Ciências - Introdução à Filosofia e à Ética das Ciências. Trad. de Luiz Paulo Rouanet. São Paulo: UNESP, 1995
- HUISMAN, D. Dicionário de Obras Filosóficas. Trad. de Ivone C. Benedetti. São Paulo: Martins Fontes, 2000.
- LALANDE, A. Vocabulário Técnico e Crítico da Filosofia. Trad. de Fátima Sá Correia, Maria E.V. Aguiar, José E. Torres e Maria G. de Souza. São Paulo: Martins Fontes, 1999.

Dentro da Linguagem de Modelagem Unificada , Material da Rational

UML destilada: Aplicando o padrão de linguagem de modelagem de objeto , Martin Fowler

Curso on-line da TogetherSoft , [practical_guides / umlonlinecourse /
www.togethersoft.com/services/](http://practical_guides/umlonlinecourse/www.togethersoft.com/services/)

Especificação da Linguagem UML Versão 1.4 , OMG

Arquitetura de Software e UML , Grady Booch (Seminário).